논문 19-44-11-19 The Journal of Korean Institute of Communications and Information Sciences '19-11 Vol.44 No.11
https://doi.org/10.7840/kics.2019.44.11.2123

기댓값 최대화 알고리즘을 적용한 네트워크 임베딩 기반 링크 예측

박 경 택•, 트 란 콩*, 신 원 용°

Network-Embedding-Based Link Prediction Using the Expectation Maximization Algorithm

Gyeong-Taek Park•, Cong Tran*,
Won-Yong Shin°

요 약

최근 네트워크 연구 분야에서 딥러닝을 적용한 네트워크 임베딩 기법이 소개되었는데, 노드 분류, 링크 예측 등의 다운스트림 기계학습 문제를 해결하는 데 있어 기존의 방법보다 효과적인 성능을 보이는 것으로 알려져 있다. 본 레터에서는 이 중 링크 예측 정확도를 개선하기 위해 기댓값 최대화 알고리즘을 네트워크 임베딩에 적용하는 방법을 제안한다. AUC 측면에서 제안한 방법의 우수성을 검증한다.

Key Words : AUC, Expectation maximization (EM) algorithm, Link prediction, Machine learning, Network embedding

ABSTRACT

Recently, network embedding methods employing deep learning have been introduced in the field of network science, and are shown to perform more effectively than conventional methods in the sense of solving downstream machine learning tasks such as node classification, link prediction, and so forth. In this letter, we propose a method that applies the expectation maximization (EM) algorithm to network embedding to improve the accuracy of link prediction. The superiority of the proposed method is shown in terms of AUC.

Ⅰ. 서 론

기존의 네트워크 연구 분야에서는 네트워크를 분석하기 위하여 네트워크 위상 (topology)에 기반한 분석 방법과 그래프 임베딩 방법, 이렇게 크게 두 가지 방법이 쓰여왔다. 네트워크 위상에 기반한 분석은 네트워크의 인접 행렬로부터 직접적으로, 그래프 임베딩 방법은 인접 행렬의 차원 축소를 통해 네트워크 분야의 다양한 다운스트림 기계 학습 추론 문제를 풀어낸다. 그러나 전자는 불필요한 정보나 잡음을 포함한다는 점에서, 후자는 수학적인 대수 구조가 실제 세계의 네트워크를 잘 대표하지 못한다는 점에서 한계를 지닌다[1].

최근 딥러닝의 발전에 힘입어 네트워크 연구 분야에서도 약진이 이루어졌는데, 네트워크 임베딩 기술이 바로 그것이다. 네트워크 임베딩이란 노드 분류, 링크 예측 등 다운스트림 기계 학습 응용을 수행하기 위해 네트워크 위상 정보를 임베딩 공간에 사상시키는 벡터화 과정을 말한다. 예를 들어 특정 소셜 미디어 내에서 사용자들의 연결 관계가 링크로 매핑된다고 가정하면, 관계망 정보로부터 비슷한 취향을 가진 사람들을 군집화하거나 연결 관계가 있을 법한 새로운 링크를 찾아주는 등의 일이 네트워크 임베딩이라는 강력한 무기를 통해 더 효과적으로 가능해진다. 네트워크 임베딩 기법은 자연어 처리 분야에서 제안된 워드 임베딩 (word embedding)[2] 아이디어를 적용한 네트워크 임베딩 기법인 DeepWalk[3]를 시작으로 다양한 기법들이 연구되어 오고 있다[4-10].

이들 기법은 취하는 방식에 따라 다음의 몇 가지 대표적인 예시를 들 수 있다. 먼저, 주어진 네트워크 구조에서 임의로 노드 시퀀스들을 추출하는 랜덤 워

※ 본 연구는 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2017R1D1A1A09000835)임.
• First Author : (ORCID:0000-0002-1843-2763)Department of Mechanical Engineering, Yonsei University, afterthougt@ gmail.com, 학생회원
° Corresponding Author : (ORCID:0000-0002-6533-3469)Department of Computational Science and Engineering, Yonsei University, wyshin@dankook.ac.kr, 종신회원
* (ORCID:0000-0001-9467-4978)Department of Computer Science and Engineering, Dankook University & Department of Computational Science and Engineering, Yonsei University, trancong208@gmail.com, 학생회원
논문번호 : 201907-123-B-LU, Received July 2 28, 2019; Revised September 7, 2019; Accepted September 16, 2019

크 (random walk) 방법을 사용하여 각 노드 시퀀스에 워드 임베딩을 적용하는 기술은 DeepWalk[3]와 node2vec[4]이 대표적이다. 다음으로, 네트워크 내 노드 간의 인접한 정도 (proximity)를 살피고 어떤 노드에 연결된 주변부 노드 (context)를 통해 그 노드의 성질을 파악하는 기술은 LINE[5]과 GraRep[6] 등이 있다.

상기 네트워크 임베딩 기술은 모두 궁극적으로는 다운스트림 기계 학습 응용 문제를 풀어내기 위한 전처리 과정으로써 기존의 네트워크 분야에서 인접 행렬을 사용하는 네트워크 모델링 방식과는 달리, 주어진 네트워크 구조를 저차원의 벡터 공간으로 효과적으로 매핑할 수 있다는 점에서 시간 및 메모리 공간 측면에서 이점이 상당하다. 본 레터에서는 위에서 기술한 다운스트림 기계 학습 응용 중 주어진 네트워크 구조로부터 잠재적 링크를 추론해내는 링크 예측 문제에 초점을 맞춘다.

한편, 잠재 정보를 추론하는 문제를 푸는 데 쓰이는 통계 기법으로 기댓값 최대화 알고리즘[11]이 있다. 기댓값 최대화 알고리즘이란 주어진 데이터가 관찰 가능하고 이와 밀접하게 관련된 어떤 정보가 관찰되지 않아 잠재적인 경우 이 정보의 가장 그럴듯한 모델을 추정 시 사용하는 반복 알고리즘이다. 기댓값 최대화 알고리즘의 각 반복 과정은 E-step과 M-step으로 나뉘는데, E-step에서는 잠재 정보에 대한 사후 분포의 국소적인 하한을 계산하고 M-step에서는 이 하한을 최적화하여 갱신하게 된다.

본 레터에서는 네트워크 임베딩에서 정의된 링크 예측 변수를 사용하여 기댓값 최대화 알고리즘의 도움을 통해 링크 존재 여부의 예측 정확도를 개선시킨 방법을 제안한다. 제안한 방법의 개선 정도를 분석하기 위해 모든 분류 임곗값에서 성능을 측정하는 그래프 ROC (receiver operating charateristic)의 아래 면적인 AUC (area under curve)를 사용한다. 여기에서 ROC는 TPR (true positive rate)과 FPR (false positive rate)간의 상관관계를 나타낸 그래프로, 이 때 TPR은 actual positive에 대한 true positive의 비율을, FPR은 actual negative에 대한 false positive의 비율을 의미한다. 실험 결과로써 임베딩 샘플 수와 반복 시행 횟수가 증가함에 따라 AUC 측면에서 링크 예측 성능이 향상됨을 확인한다.

Ⅱ. 방법론

주어진 정적 네트워크 $G=(V,E)$의 구조 정보로부터 잠재 링크를 예측하는 다운스트림 기계학습 문제를 고려한다. 네트워크 임베딩 모델의 학습을 위해 주어진 노드 구성으로부터 가능한 모든 엣지들을 학습 집합과 테스트 집합으로 나눈다. 이때 잠재 링크 존재 여부는 테스트 집합 내 링크에 각각 부여된 확률로 확인 가능하다.

우리는 임의의 두 노드 사이에 링크가 존재할 확률을 원소로 갖는 행렬을 구성하고 이를 링크 예측 변수 $\phi \in R^{|V|\times|V|}$라 한다. 그림 1의 상단 초기화 세팅에서 확인할 수 있듯이, 초기 링크 예측 변수를 정의하기 위해 네트워크 $G=(V,E)$의 인접 행렬이 네트워크 임베딩을 1회 거친 출력을 사용하는데, 여기서 임베딩은 노드 임베딩과 엣지 임베딩, 그리고 링크 예측과 같은 분류 문제 해결을 위한 기계학습 방법인 로지스틱 회귀 (logistic regression) 순서로 구성된다. 각 반복 시행에서 N회의 독립적인 샘플링을 수행하는데, 샘플링 후 임베딩 함수의 입력은 링크 예측 변수 ϕ의 각 원소에 대하여 베르누이 시행을 통해 얻은 인접 행렬을 사용한다. 여기서 $i\,(1\le i\le N)$번째 샘플의 임베딩에 대응되는 입력 인접 행렬을 $a_\phi^{(i)}$, i번째 샘플의 임베딩을 거친 테스트 집합 내 링크들의 존재 추정 확률값의 집합을 $z^{(i)}$라 정의하고, 각 반복 시행에서 다음의 우도를 최대화하는 최적화 문제를 고려한다.

$$\sum_{i=1}^{N}\log p(a_\phi^{(i)};\phi) \tag{1}$$

기댓값 최대화 알고리즘을 적용하여 위의 로그 우도를 최대화하는 대신 (1)의 하한을 최대화하는 방법을 취한다. $z^{(i)}$가 특정 분포 $Q_i(z^{(i)})$를 가진다고 할 때 (1)은 다음과 같이 표현될 수 있다.

$$\begin{aligned}
&\sum_{i=1}^{N}\log p(a_\phi^{(i)};\phi)\\
&=\sum_{i=1}^{N}\log\sum_{z^{(i)}}p(a_\phi^{(i)},z^{(i)};\phi)\\
&=\sum_{i=1}^{N}\log\sum_{z^{(i)}}Q_i(z^{(i)})\frac{p(a_\phi^{(i)},z^{(i)};\phi)}{Q_i(z^{(i)})}\\
&\ge\sum_{i=1}^{N}\sum_{z^{(i)}}Q_i(z^{(i)})\log\frac{p(a_\phi^{(i)},z^{(i)};\phi)}{Q_i(z^{(i)})}.
\end{aligned} \tag{2}$$

이 때, 부등식은 Jensen 부등식을 따른다. $Q_i(z^{(i)})=p(z^{(i)}|a_\phi^{(i)};\phi)$로 두면 (2)의 하한은 다음과 같이 정리할 수 있다.

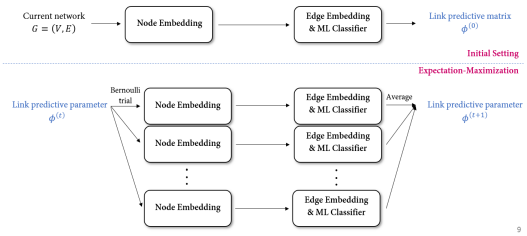

그림 1. 제안한 알고리즘의 블록도
Fig. 1. Block diagram of the proposed algorithm

$$\sum_{i=1}^{N}\sum_{z^{(i)}} Q_i(z^{(i)})\log\frac{p(a_\phi^{(i)}, z^{(i)};\phi)}{Q_i(z^{(i)})}$$
$$= \sum_{i=1}^{N}\sum_{z^{(i)}} p(z^{(i)}|a_\phi^{(i)};\phi)\log\frac{p(a_\phi^{(i)}, z^{(i)};\phi)}{p(z^{(i)}|a_\phi^{(i)};\phi)}$$
$$= \sum_{i=1}^{N}\sum_{z^{(i)}} p(z^{(i)}|a_\phi^{(i)};\phi)\log p(a_\phi^{(i)};\phi).$$

따라서 기댓값 최대화 알고리즘의 적용 시 각 단계는 다음과 같이 요약된다.

$(E-step)$ f를 계산한다.

$$f(\phi|\phi^{(t)}) := E_{z^{(i)}\sim Q_i}[\log p(a_\phi^{(i)};\phi)].$$

$(M-step)$ ϕ를 갱신한다.

$$\phi^{(t+1)} = \arg\max_\phi f(\phi|\phi^{(t)}).$$

E-step에서는 이전 반복 시행 $\phi^{(t)}$로부터 얻은 N회의 샘플링 후 임베딩을 사용하여 잠재 링크의 존재 확률을 추정하며, M-step에서는 다음 반복 시행 $\phi^{(t+1)}$에서 링크 예측을 수행하기 위하여 링크 예측 파라미터 ϕ를 갱신한다. 이는 그림 1의 하단 기댓값 최대화 과정에서 확인할 수 있다.

Ⅲ. 실 험

링크 예측 성능 측정을 위한 데이터셋으로는 Karate network와 Dolphins network를 사용한다. 각각의 네트워크에서 두 노드 사이에 존재하는 링크를 positive edge, 그렇지 않은 링크를 negative edge라 하고 각각의 엣지 집합을 학습 집합과 테스트 집합의 비율이 8:2가 되도록 데이터셋을 구성한다.

노드 임베딩 방법으로는 node2vec[5] 기법을 사용하는데, 하이퍼파라미터 (hyperparameter)의 정의를 통해 random walk 방법에 bias를 가하여 주어진 네트워크에서 노드 시퀀스를 추출하고, 여기에 워드 임베딩[2] 방법을 적용한 기법이다. 주요한 하이퍼파라미터로써 노드의 특징 벡터의 차원 $d = 16$, random walk의 길이 $l = 10$, 각 노드 당 random walk 수 $r = 20$을 사용하였다.

성능 측정을 위한 척도로 ROC 곡선 아래 면적인 AUC를 이용한다. 그림 2와 그림 3은 각각 Karate network와 Dolphins network에 대해 반복 시행 횟수가 각각 1, 3, 5일 때, 샘플 수의 증가에 따른 AUC를 보여준다. 그 결과, 반복 시행 횟수의 증가 및 샘플 수의 증가에 따라 링크 예측 성능이 기댓값 최대화 알고리즘을 적용하지 않은 기존 node2vec 방법 (Initial) 대비 최대 5% 정도 향상되는 것을 볼 수 있으며, 특정 반복 시행 횟수 및 특정 샘플 임베딩 수 이후에 포화 현상이 일어남을 확인하였다.

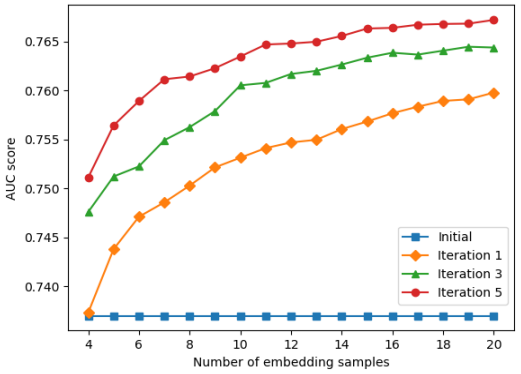

그림 2. Karate network에서 임베딩 샘플 수에 따른 AUC score
Fig. 2. AUC score according to the number of embedding samples in the Karate network

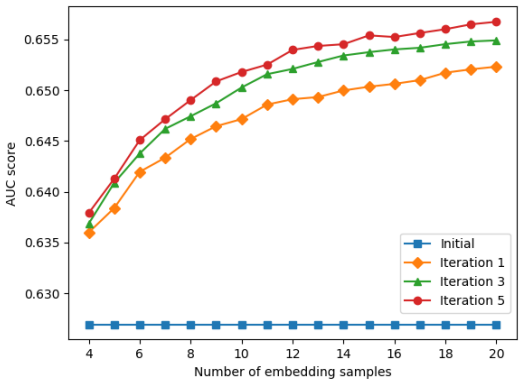

그림 3. Dolphins network에서 임베딩 샘플 수에 따른 AUC score
Fig. 3. AUC score according to the number of embedding samples in the Dolphins network

Ⅳ. 결 론

본 레터에서는 네트워크 임베딩 기법에 기댓값 최대화 알고리즘을 적용하여 링크 예측의 성능을 높이는 방법을 제안하였다. Karate 및 Dolphins network에서 링크 예측 실험 결과, 제안한 방법은 추가적인 사전 정보 없이도 네트워크 임베딩을 사용한 링크 예측 성능을 개선하는 것을 확인하였다.

References

[1] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowledge and Data Eng.*, vol. 31, no. 5, pp. 833-852, Jun. 2018.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.,* Lake Tahoe, Nevada, Dec. 2013.

[3] B. Perrozi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining,* New York, USA, Aug. 2014.

[4] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 855-864, San Francisco, CA, USA, Aug. 2016.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015.

[6] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. and Knowledge Management,* pp. 891-900, New York, NY, USA, Oct. 2015.

[7] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining,* pp. 1225-1234, San Francisco, CA, USA, Aug. 2016.

[8] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining,* pp. 1105-1114, San Francisco, CA, USA, Aug. 2016.

[9] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.,* pp. 1025-1035, Long Beach, CA, USA, Dec. 2017.

[10] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," *ArXiv* abs/1801. 10247 (2018).

[11] T. K. Moon, "The expectation- maximization algorithm," *IEEE Sign. Process. Mag.*, vol. 13, no. 6, pp. 47-60, Nov. 1996.